

Sun Azimuth and Elevation Algorithms

DRAFT

**D. Crehan
R. McIntosh**

January 23, 1995

This report describes and analyzes a Sun azimuth and elevation computation algorithm intended for implementation in the Landsat Processing System (LPS). The report provides the following information.

- A summary of the algorithm's input data and their sources.
- A description of the algorithm.
- An analysis of the algorithm's computational complexity.
- An estimate of the Delivered Source Instructions (DSIs) required to implement the algorithm.
- Source listings for FORTRAN subroutines that compute the Geocentric Inertial (GCI) Sun vector and Greenwich Hour Angle (GHA).

Input Data and Their Sources

The Sun azimuth and elevation computation algorithm requires the inputs listed, along with their sources, in Table 1.

- Latitude (*lat*) and longitude (*lon*) of the WRS scene center.
- Spacecraft time (*t*) at which the WRS scene center datum was obtained.

All values are computed during WRS scene identification.

Algorithm Description

The problem to be addressed is this. Given the latitude (*lat*), longitude (*lon*), and spacecraft time (*t*) of the WRS scene center, compute the Sun azimuth (AZ) and elevation (EL) at the ground point. The algorithm is as follows.

1. Compute the Greenwich Hour Angle (GHA, right ascension of Greenwich) at time *t* using subroutine JGHAX (attached).
2. Compute the GCI Sun vector (\vec{R}_{Sur}) at time *t* using subroutine SOL (attached).
3. Define an Earth-fixed coordinate system centered at the latitude (*lat*) and longitude (*lon*) of interest with coordinate axes pointing north (N), east (E), and local vertical (V)

$$\hat{V} = \begin{vmatrix} \cos(\text{lon}) \cos(\text{lat}) \\ \sin(\text{lon}) \cos(\text{lat}) \\ \sin(\text{lat}) \end{vmatrix}$$

$$\hat{E} = \begin{vmatrix} -\sin(\text{lon}) \\ \cos(\text{lon}) \\ 0 \end{vmatrix}$$

$$\hat{N} = \begin{vmatrix} -\cos(\text{lon}) \sin(\text{lat}) \\ -\sin(\text{lon}) \sin(\text{lat}) \\ \cos(\text{lat}) \end{vmatrix}$$

4. Compute the transformation matrix from Earth-fixed to GCI

$$[G] = \begin{vmatrix} \cos(\text{GHA}) & -\sin(\text{GHA}) & 0 \\ \sin(\text{GHA}) & \cos(\text{GHA}) & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

5. Transform the Earth-fixed coordinate axes to inertial

$$\hat{V}_N = [G]\hat{N}$$

$$\hat{V}_E = [G]\hat{E}$$

$$\hat{V}_V = [G]\hat{V}$$

6. Compute the azimuth and elevation (AZ, EL) of the Sun.

$$AZ = \arctan \frac{\hat{V}_N \cdot \hat{R}_{Sun}}{\hat{V}_E \cdot \hat{R}_{Sun}}$$

$$EL = \arcsin(\hat{V}_V \cdot \hat{R}_{Sun})$$

Computational Complexity

The algorithm is of constant complexity ("O(C)"). All computations are double-precision floating point, and their cost will dominate. Table 3 indicates the type and number of floating point operations required to compute the Sun azimuth and elevation of a single datum. GCI Sun Vector and GHA computation complexities are listed separately.

Table 3: Floating Point Operations for WRS Scene ID Algorithms

Algorithm	+/-	-N	x	÷	MOD	Roun d	Trig
Azimuth/ Elevation	24	4	36	1	0	0	18
GCI Sun Vector	7	0	7	0	1	0	5
GHA	579	4	560	5	1	1	45
Total	610	8	603	6	2	1	20

Assuming 1 computation for the scene center per scene $1,227 + 20n$ (n = the mean number of floating point operations to compute a trigonometric function) floating point operations will be required for each scene. At 250 scenes per day, approximately $306,750 + 5,000n$ floating point operations per day will be required

DSI Estimates

Table 4 presents DSI estimates for each of the algorithms. For GCI Sun Vector and GHA computations, the DSI estimate is a count of the DSI in the existing FORTRAN implementations. For the top-level algorithm, the estimate is based on the number of floating point operations required (assuming 1 DSI per operation), the number of distinct intermediate values computed (assuming 1 DSI per value), a 30% overhead for initialization and termination handling, iteration, etc., and 100% overhead for exception handling.

Table 4: DSI Estimates for WRS Scene Identification Algorithms

Algorithm	Estimated DSI
Sun Azimuth/Elevation at Spacecraft	198
Sun Azimuth/Elevation at Earth	226
GCI Sun Vector	22
GHA	218
Total	664

Sun Azimuth and Elevation Algorithm Issues and Questions

1. Time conversions required (e.g. from UTC to A.1) have not been incorporated into the algorithm.
2. Supplying the A.1-UT1 time difference to subroutine JGHAX, which computes the GHA, has not been incorporated into the algorithm. A direct transformation from UTC to UT1 may be more appropriate to the LPS version of this subroutine.

GCI Sun Vector Subroutine in FORTRAN

```

C           DATA SET SOL          AT LEVEL 001 AS OF 01/23/86
C***** ****
C*
C*   NAME:
C*     SOL
C*
C*   PURPOSE:
C*     TO DETERMINE THE POSITION OF THE SUN WITH RESPECT TO C*      THE
EARTH
C*
C*   COMPUTER:
C*     NAS R1/R2 MVS OPERATING SYSTEM
C*
C*   LANGUAGE:
C*     VS FORTRAN
C*
C*   METHOD:
C*     POSITION OF THE SUN WITH RESPECT TO THE EARTH IS
C*     DETERMINED FROM THE MEAN MOTION OF THE SUN AS
C*     DESCRIBED IN THE SUPPLEMENT TO THE NAUTICAL
C*     EPHEMERIS SOLAR POSITION AND VELOCITY ROUTINE
C*
C*     POSITIONAL ACCURACY ABOUT .0005 RADIAN OR 75000 KM
C*
C*     FORMULAE ARE FROM THE EXPLANATORY SUPPLEMENT TO C*THE
C*     ASTRONOMICAL EPHEMERIS AND THE AMERICAN EPHEMERIS AND NAUTICAL
C*     NAUTICAL ALMANAC (1961), PAGE 98.
C*
C*   REFERENCE :
C*     CSC/TM-77/6095, "SUBROUTINE DESCRIPTION AND MATHEMATICAL MODEL
C*     FOR THE OSMEAN MEAN ELEMENTS SUBROUTINE", D.L RICHARDSON AND
C*     D. W. DUNHAM, APRIL, 1977, APPENDIX D.
C*
C*   CALLING SEQUENCE
C*     CALL SOL( AJD, X1,Y1, Z1, AL1)
C*
C*   ARGUMENT LIST
C*     NAME      TYPE    USE      DESCRIPTION
C*     ===      ===      ===      =====
C*     AJD       R*8     I       FULL JULIAN EPHEMERIS DATE
C*     X1        R*8     O       X-COMPONENT OF GEOCENTRIC SOLAR POSITION
C*                           VECTOR = X/R
C*     Y1        R*8     O       Y-COMPONENT OF GEOCENTRIC SOLAR POSITION
C*                           VECTOR = Y/R
C*     Z1        R*8     O       Z-COMPONENT OF GEOCENTRIC SOLAR POSITION
C*                           VECTOR = Z/R
C*     AL1       R*8     O       SCALING FACTOR FOR GEOCENTRIC SOLAR
C*                           DISTANCE
C*
C*   EXTERNAL VARIABLES:
C*     NAME      TYPE    USE      DESCRIPTION
C*     ===      ===      ===      =====

```

```

C*      COMMON BLOCK:  CONSTA
C*          TWOPI     R*8      I      VALUE OF 2 * PI
C*          DEG2RD   R*8      I      DEGREES TO RADIANS CONVERSION FACTOR
C*
C*
C*      EXTERNAL REFERENCES:
C*          NONE
C*
C*      I/O PROCESSING:
C*          NONE
C*
C*      ERROR PROCESSING:
C*          NONE
C*
C*      RESTRICTIONS:
C*          NONE
C*
C*
C*      HISTORY
C*          NAME          DATE      COMMENTS
C*          =====        =====    =====
C*          D. DUNHAM     12/76    ORIGINATOR
C*          B. JEFFE      4/85     UPDATED TO VSFORT, ADDED PROLOG
C*
C*****SUBROUTINE SOL(AJD, X1, Y1, Z1, AL1)
C
C      IMPLICIT REAL*8 (A - H, O - Z)
C
C      COMMON /CONSTA/RAD, PI, TWOPI, DEG2RD
C      DATA ISTART /0/
C
C***** FOLLOWING ARE FOR BENCHMARK RUNS W/OLD CONSTANTS - REMOVE**
C
C      TWOPI = 6.2831853071796D0
C      DEG2RD = .17453292519943D-01
C
C      IF (ISTART .EQ. 0) THEN
C
C      INITIALIZE CONSTANTS AND RESET ISTART VARIABLE ON 1ST ENTRY ONLY
C
C          ISTART = 1
C          GAM0 = 281.220833D0 * DEG2RD
C          GAMD = .0000470684D0 * DEG2RD
C          G0 = 358.475845D0 * DEG2RD
C          GD = 0.985600267D0 * DEG2RD
C          EPS0 = 23.452294D0 * DEG2RD
C          EPSSD = 0.35626D-06 * DEG2RD
C
C      ENDIF
C
C          D = AJD - 2415020.0D0                      ± E
C          E = 0.01675104D0 - 0.11444D-08*D           ± .1444E-8 * E
C          E1 = 2.D0*E
C          EPS = EPS0 - D*EPSSD
C          GAM = GAM0 + GAMD*D
C          G = G0 + D*GD
C          AL = GAM + G + E1*DSIN (G)

```

```
AL = DMOD (AL, TWOPI)
X1 = DCOS (AL)
Z1 = DSIN (AL)
Y1 = Z1*DCOS (EPS)
Z1 = Z1*DSIN (EPS)
AL1 = E*DCOS (G)

C
C***** FOLLOWING RESETS TO OLD CONSTANTS IN COMMON - REMOVE*****
C
C      TWOPI = 6.2831853D0
C      DEG2RD = .01745329252D0
C
C      RETURN
C      END
```

Greenwich Hour Angle (GHA) Subroutine in FORTRAN

```

C/ ADD NAME=UTJGHAX                               04MAY90 15.38.35
C          DATA SET UTJGHAX      AT LEVEL 003 AS OF 10/30/89
C          DATA SET UTJGHAX      AT LEVEL 003 AS OF 01/31/89
C          DATA SET UTJGHAX      AT LEVEL 008 AS OF 11/16/88
C          DATA SET UTJGHAX      AT LEVEL 001 AS OF 07/21/88
C          DATA SET UTJGHAX      AT LEVEL 001 AS OF 06/20/88
C          SUBROUTINE JGHAX(XJDA1,A1MUT1,GHA)
CC
CC PURPOSE: COMPUTES GREENWICH HOUR ANGLE USING ANALYTICAL DATA
CC
CC METHOD:  GREENWICH HOUR ANGLE IS COMPUTED AS THE SUM OF THE
CC           GREENWICH MEAN SIDEREAL TIME AT 0.0H OF DAY, THE
CC           UNIVERSAL TIME AND THE NUTATION CORRECTION TERM.
CC
CC ARGUMENT LIST:
CC   ARGUMENT  TYPE   I/O  DESCRIPTION <DIM>
CC   XJDA1    R8     I    A.1 JULIAN DATE
CC   A1MUT1   R8     I    A.1-UT1 DIFFERENCE (SEC)
CC   GHA       R8     O    GREENWICH HOUR ANGLE (RAD)
CC
CC CALLING SUBROUTINE: USER ROUTINE
CC
CC SUBROUTINES CALLED: JNUTON
CC
CC COMMON BLOCK VARIABLES USED: NONE
CC
CC MODIFICATIONS
CC NAME          DATE      DESCRIPTION
CC J. ROITZ      10/21/88  1) ADD A1-UT1 DIFFERENCE TO THE
CC                           ARGUMENT LIST AS UT1 IS NEEDED TO
CC                           COMPUTE GMST
CC                           2) CORRECTLY DETERMINE THE NUTATION
CC                           CORRECTION
CC
CC -DECLARE EXTERNAL VARIABLES
CC   REAL*8 XJDA1,A1MUT1,GHA
CC -DECLARE LOCAL VARIABLES
CC   REAL*8 C0,C1,C2,C3,TWOP1,ROTAT,GHA1,GHA2,GHA3,UT1,UTNOON,UTMID,
$ ANUT(3,3), CNT, DLONG, TROBL, CTROBL
CC   DATA C0      / 1.7533685592332653D0 /
CC   DATA C1      / 628.33197068884084D0 /
CC   DATA C2      / 0.67707139449033354D-05 /
CC   DATA C3      / -0.45087672343186841D-09 /
CC   DATA TWOP1   / 6.2831853071795865D0 /
CC   DATA ROTAT   / 7.2921151467D-05 /
CC
CC -DETERMINE UT1
CC
CC   UT1 = XJDA1 - A1MUT1/86400.0D0
CC
CC -COMPUTE NUMBER OF CENTURIES FROM J2000.0
CC

```

```

CC      FUNCTION DNINT ROUNDS TO THE NEAREST WHOLE NUMBER, THUS
CC      NOON OF THE CURRENT DAY.  SUBTRACTING HALF A DAY RESULTS IN
CC      0 HR OF THE CURRENT DAY.

CC      UTMNOON=DNINT(UT1)
CC      UTMID=UTNOON-0.5D0
CC      CNT=(UTMID-2451545.D0)/36525.D0
CC      -COMPUTE GREENWICH MEAN SIDEREAL ANGLE
CC      GHA1=C0+(C1+(C2+C3*CNT)*CNT)*CNT
CC      GHA2=ROTAT*86400.D0*(UT1-UTMID)
CC      -COMPUTE NUTATION CORRECTION
CC      THE NUTATION CORRECTION, GHA3, IS DEFINED AS THE PRODUCT OF
CC      COSINE OF TRUE OBLIQUITY AND NUTATION IN LONGITUDE.
CC      IN THE NUTATION MATRIX A,
CC      A(2,1) IS THE PRODUCT OF THE COSINE OF TRUE OBLIQUITY AND
CC      THE SINE OF THE NUTATION IN LONGITUDE.
CC      A(3,1) IS THE PRODUCT OF THE SINE OF TRUE OBLIQUITY AND
CC      THE SINE OF THE NUTATION IN LONGITUDE.
CC      A(3,1)/A(2,1) IS THE TANGENT OF THE TRUE OBLIQUITY.
CC      THESE WILL BE USED TO DETERMINE THE NUTATION IN LONGITUDE
CC      AND THEN GHA3
CC
CC      CALL JNUTON(XJDA1,ANUT,0,0)
CC      TROBL = DATAN2(ANUT(3,1),ANUT(2,1))
CC      CTROBL= DCOS(TROBL)
CC      DLONG = DASIN(ANUT(2,1)/CTROBL)
CC      GHA3 = DLONG*CTROBL
CC
CC      -ADD NUTATION CORRECTION TO MEAN VALUE AND SET POSITIVE < 2 PI
CC      GHA=DMOD(GHA1+GHA2+GHA3,TWOPi)
CC      IF(GHA.LT.0.D0)GHA=GHA+TWOPi
CC      RETURN
CC      END
C/ ADD NAME=UTJNUTON          04MAY90 15.38.35
C      DATA SET UTJNUTON    AT LEVEL 003 AS OF 10/30/89
C      DATA SET UTJNUTON    AT LEVEL 002 AS OF 08/03/88
C      DATA SET UTJNUTON    AT LEVEL 001 AS OF 07/21/88
C      DATA SET UTJNUTON    AT LEVEL 001 AS OF 09/30/87
CCCC
C
C      SUBROUTINE JNUTON (T2JA1, ANUT, IFDBG, IUDBG)
C
C      1 FORMAT(' **** SUBROUTINE JNUTON   SEPTEMBER 15, 1987 ****')
C
C-----LANGUAGE - VS FORTRAN
C
C-----FUNCTION - JNUTON COMPUTES THE NUTATION OF THE EARTH
C                  AT TIME T2JA1 USING THE IAU 1980 THEORY OF NUTATION
C                  EQUATIONS FOR THE MEAN-OF-J2000 COORDINATE SYSTEM.  THE
C                  OUTPUT IS THE MEAN-EQUATOR-OF-DATE TO TRUE-EQUATOR-OF-
C                  DATE ROTATION MATRIX.
C
C-----MATHEMATICAL METHOD - REFER TO THE SUPPLEMENT TO THE ASTRONOMICAL
C                  ALMANAC FOR 1984, PAGES S15 AND S23-S26.
C                  THIS SUBROUTINE IS A TRUNCATED VERSION OF
C                  THE NUTATION ALGORITHM. IT INCLUDES THE
C                  FIRST SECULAR TERM FOR EACH SERIES AND ALL
C                  TERMS WITH A COEFFICIENT OF 0.001 ARCSECOND

```

C OR GREATER IN MAGNITUDE (TERMS 1-4, 9-19,
C AND 31-50).
C THE CORRECT TIME INPUT TO THE ALGORITHM IS
C BARYCENTRIC DYNAMICAL TIME (TDB). THE TDT
C (OR EPHemeris TIME) TO TDB CORRECTION IS
C PERIODIC WITH THE AMPLITUDE LESS THAN
C 0.002 SECONDS.
C THE THREE ROTATION ANGLES ARE:
C 1) MEAN OBLIQUITY OF THE ECLIPTIC
C (MEAN-EQUATOR- TO MEAN-ECLIPTIC-OF-DATE)
C 2) NEGATIVE NUTATION IN LONGITUDE
C 3) NEGATIVE TRUE OBLIQUITY OF THE ECLIPTIC
C (TRUE-ECLIPTIC- TO TRUE-EQUATOR-OF-DATE)
C
C-----ARGUMENTS -
C ARGUMENT TYPE IO DESCRIPTION
C ----- --
C T2JA1 R*8 I A.1 JULIAN DATE OF THE EPOCH IN DAYS PLUS
C FRACTION OF DAY
C ANUT(3,3) R*8 O ROTATION MATRIX FOR NUTATION FOR T2JA1
C (MEAN-EQUATOR-OF-DATE TO TRUE-EQUATOR-OF-DATE)
C IFDBG I*4 I FLAG TO REQUEST DEBUG PRINTER OUTPUT
C = 0 ... NO PRINTER OUTPUT
C = 1 ... PRINT INPUT AND OUTPUT
C = 2 ... PRINT INPUT, OUTPUT, CONSTANTS
C IUDBG I*4 I FORTRAN UNIT NUMBER FOR DEBUG PRINTER OUTPUT
C
C-----EXTERNAL REFERENCES -
C JOBLTY - COMPUTES THE MEAN OBLIQUITY OR TRUE OBLIQUITY AND THE
C ECLIPTIC TO EQUATOR ROTATION MATRIX
C
C-----CALLED BY - JPREFNU OR USER ROUTINE
C
C-----COMMONS REFERENCED - JCOEF
C-----VARIABLES -
C VARIABLE TYPE DESCRIPTION
C -----
C TJDSTD R*8 STANDARD EPOCH FOR THE J2000 COORDINATE SYSTEM
C IN DAYS PLUS FRACTION OF DAY
C DAYCEN R*8 NUMBER OF DAYS IN A JULIAN CENTURY
C DA1TDT R*8 A.1 ATOMIC TIME TO TERRESTRIAL DYNAMICAL TIME
C (TDT) DIFFERENCE IN DAYS
C ARCTR R*8 ARCSECONDS TO RADIAN CONVERSION FACTOR
C CFARGS R*8 COEFFICIENTS FOR COMPUTING THE FUNDAMENTAL
C ARGUMENTS OF THE NUTATION SERIES IN ARCSECONDS
C CFMULT R*8 COEFFICIENTS OF THE FUNDAMENTAL ARGUMENTS FOR
C THE NUTATION SERIES IN ARCSECONDS
C CFDLON R*8 COEFFICIENTS OF THE LONGITUDE NUTATION TERMS
C FOR THE NUTATION SERIES IN ARCSECONDS
C CFTDLN R*8 COEFFICIENTS OF THE SECULAR LONGITUDE NUTATION
C TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C NTERMS I*4 THE NUMBER OF NONSECULAR TERMS INCLUDED IN THE
C COMPUTATION
C NSECL I*4 THE NUMBER OF SECULAR TERMS INCLUDED IN THE
C COMPUTATION
C
C-----ERROR HANDLING - NONE

```

C-----FILES REFERENCED - NONE
C
C-----DESIGNER - S. DEVLIN, CSC, AUGUST 20, 1987
C
C-----PROGRAMMER - S. DEVLIN, CSC SEPTEMBER 15, 1987
C
C-----VERIFIED BY -
C
C-----MODIFICATIONS -
C      NAME      DATE          DESCRIPTION
C
CCCC
C      -DECLARE ARGUMENTS
      REAL*8 T2JA1
      REAL*8 ANUT(3,3)
      INTEGER*4 IFDBG
      INTEGER*4 IUDBG
C      -DECLARE LOCAL VARIABLES
      REAL*8 TREF
      REAL*8 ARGS(5)
      REAL*8 ANGL, DLONG
      REAL*8 AMOB(3,3), ATOB(3,3)
      REAL*8 COBLTM, SOBLTM, CTROBL, STROBL, CDLONG, SDLONG
      INTEGER*4 I, J, IOPT
C
C      -COMMON BLOCK JCOEF VARIABLES
      REAL*8 TJDSTD
      REAL*8 DAYCEN, DA1TDT, ARCTR
      REAL*8 CFZETA(6), CFZEE(6), CFTHET(6)
      REAL*8 CFOBLM(4)
      REAL*8 CFARGS(4,5)
      REAL*8 CFMULT(5,35), CFDLON(35), CFDOBL(35)
      REAL*8 CFTDLN(1), CFTDOB(1)
      INTEGER*4 NTERMS
      INTEGER*4 NSECL
C
C      COMMON /JCOEF/ TJDSTD, DAYCEN, DA1TDT, ARCTR,
2           CFZETA, CFZEE, CFTHET,
3           CFOBLM, CFARGS, CFMULT, CFDLON, CFDOBL,
4           CFTDLN, CFTDOB, NTERMS, NSECL
C
C      PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
          WRITE(IUDBG,1)
          WRITE(IUDBG,1000) T2JA1
          IF(IFDBG.GT.1) THEN
              WRITE(IUDBG,2003) TJDSTD, DAYCEN, DA1TDT, ARCTR
              WRITE(IUDBG,2004) NTERMS, NSECL, CFTDLN
              WRITE(IUDBG,2006) (I,(CFMULT(J,I),J=1,5),CFDLON(I),I=1,NTERMS)
          END IF
      END IF
C
C      COMPUTE THE TIME FROM J2000 TO THE EPOCH TIME IN CENTURIES
      TREF = ((T2JA1+DA1TDT) - TJDSTD) / DAYCEN
C
C      COMPUTE FUNDAMENTAL ARGUMENTS FOR NUTATION SERIES
C

```

```

      DO 100 I=1,5
      ARGS(I) = ARCTR * (CFARGS(1,I) + TREF * (CFARGS(2,I) +
      2          TREF * (CFARGS(3,I) + TREF * CFARGS(4,I)) ))
100  CONTINUE
C
C      COMPUTE NUTATION IN LONGITUDE
C
      DLONG = 0.0D0
      DO 300 I=1,NTERMS
C
      ANGL = 0.0D0
      DO 200 J=1,5
      ANGL = ANGL + CFMULT(J,I) * ARGS(J)
200  CONTINUE
C
      IF(I.GT.NSECL) THEN
          DLONG = DLONG + CFDLON(I) * DSIN(ANGL)
      ELSE
          DLONG = DLONG + (CFDLON(I) + CFTDLN(I)*TREF) * DSIN(ANGL)
      END IF
300  CONTINUE
      DLONG = ARCTR * DLONG
C
C      COMPUTE MEAN OBLIQUITY AND TRUE OBLIQUITY
C
      IOPT = 1
      CALL JOBLTY(T2JA1, AMOB, IOPT, IFDBG, IUDBG)
      IOPT = 2
      CALL JOBLTY(T2JA1, ATOB, IOPT, IFDBG, IUDBG)
C
C      COMPUTE THE ROTATION MATRIX
C
      CDLONG = DCOS(DLONG)
      SDLONG = DSIN(DLONG)
      COBLTM = AMOB(2,2)
      SOBLTM = AMOB(3,2)
      CTROBL = ATOB(2,2)
      STROBL = ATOB(3,2)
C
      ANUT(1,1) = CDLONG
      ANUT(1,2) = -SDLONG*COBLTM
      ANUT(1,3) = -SDLONG*SOBLTM
      ANUT(2,1) = CTROBL*SDLONG
      ANUT(2,2) = CTROBL*CDLONG*COBLTM + STROBL*SOBLTM
      ANUT(2,3) = CTROBL*CDLONG*SOBLTM - STROBL*COBLTM
      ANUT(3,1) = STROBL*SDLONG
      ANUT(3,2) = STROBL*CDLONG*COBLTM - CTROBL*SOBLTM
      ANUT(3,3) = STROBL*CDLONG*SOBLTM + CTROBL*COBLTM
C
C      PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
          WRITE(IUDBG,2001) TREF, DLONG, ARGS
          WRITE(IUDBG,2002) ((ANUT(I,J),J=1,3),I=1,3)
      END IF
C
      1000 FORMAT(' T2JA1 = ',F18.8)
      2001 FORMAT('0TREF, DLONG = ',2D24.15,/,
                 $ ' ARGS(1)= ',D22.14,/,
```

```

$ ' ARGS(2)= ',D22.14,/,
$ ' ARGS(3)= ',D22.14,/,
$ ' ARGS(4)= ',D22.14,/,
$ ' ARGS(5)= ',D22.14)
2002 FORMAT(' ANUT(1,1-3) = ',3F20.15,' ANUT(2,1-3) = ',3F20.15,
2   /' ANUT(3,1-3) = ',3F20.15)
2003 FORMAT(' TJDSTD, DAYCEN, DA1TDT= ',F18.8,F10.2,D24.15)
2004 FORMAT(' NTERMS, NSECL = ',2I4,
$ ' CFTDLN(1) = ',D22.12,/,
$ ' CFTDLN(2) = ',D22.12,/,
$ ' CFTDLN(3) = ',D22.12,/,
$ ' CFTDLN(4) = ',D22.12)
2006 FORMAT(1X,I4,5F5.0,D22.12)

C
      RETURN
      END

C/ ADD NAME=UTJOBLTY          04MAY90 15.38.35
C           DATA SET UTJOBLTY    AT LEVEL 003 AS OF 10/30/89
C           DATA SET UTJOBLTY    AT LEVEL 002 AS OF 08/03/88
C           DATA SET UTJOBLTY    AT LEVEL 001 AS OF 07/21/88
C           DATA SET UTJOBLTY    AT LEVEL 001 AS OF 09/30/87
CCCC
C
      SUBROUTINE JOBLTY (T2JA1, AOBL, IOPT, IFDBG, IUDBG)
C
      1 FORMAT(' ***** SUBROUTINE JOBLTY  SEPTEMBER 15, 1987 *****')
C
C-----LANGUAGE - VS FORTRAN
C
C-----FUNCTION - JOBLTY COMPUTES THE MEAN OBLIQUITY OF THE EQUATOR OR
C                 THE TRUE OBLIQUITY OF THE EQUATOR AT TIME T2JA1 USING
C                 THE IAU 1980 THEORY OF NUTATION EQUATIONS FOR THE
C                 MEAN-OF-J2000 COORDINATE SYSTEM.  THE OUTPUT IS THE
C                 ECLIPTIC-OF-DATE TO EQUATOR-OF-DATE ROTATION MATRIX.
C
C-----MATHEMATICAL METHOD - REFER TO THE SUPPLEMENT TO THE ASTRONOMICAL
C                           ALMANAC FOR 1984, PAGES S15 AND S23-S26.
C                           FOR THE TRUE-OF-DATE COORDINATE SYSTEM, THE
C                           COMPUTATION IS A TRUNCATED VERSION OF
C                           THE NUTATION ALGORITHM. IT INCLUDES THE
C                           FIRST SECULAR TERM FOR THE SERIES AND ALL
C                           TERMS WITH A COEFFICIENT OF 0.001 ARCSECOND
C                           OR GREATER IN MAGNITUDE (TERMS 1-4, 9-19,
C                           AND 31-50).
C                           THE CORRECT TIME INPUT TO THE ALGORITHM IS
C                           BARYCENTRIC DYNAMICAL TIME (TDB). THE TDT
C                           (OR EPHemeris TIME) TO TDB CORRECTION IS
C                           PERIODIC WITH THE AMPLITUDE LESS THAN
C                           0.002 SECONDS.
C
C-----ARGUMENTS -
C     ARGUMENT  TYPE  IO  DESCRIPTION
C     -----  ----  --  -----
C     T2JA1    R*8   I   A.1 JULIAN DATE OF THE EPOCH IN DAYS PLUS
C                           FRACTION OF DAY
C     AOBL(3,3) R*8   O   ROTATION MATRIX FOR OBLIQUITY FOR T2JA1
C     IOPT      I*4   I   FLAG TO INDICATE THE RESIRED ROTATION
C                           = 1 ... MEAN ECLIPTIC OF DATE TO MEAN

```

```

C                               EQUATOR OF DATE (MEAN OBLIQUITY)
C                               = 2 ... TRUE ECLIPTIC OF DATE TO TRUE
C                               EQUATOR OF DATE (TRUE OBLIQUITY)
C     IFDBG      I*4    I   FLAG TO REQUEST DEBUG PRINTER OUTPUT
C                           = 0 ... NO PRINTER OUTPUT
C                           = 1 ... PRINT INPUT AND OUTPUT
C                           = 2 ... PRINT INPUT, OUTPUT, CONSTANTS
C     IUDBG      I*4    I   FORTRAN UNIT NUMBER FOR DEBUG PRINTER OUTPUT
C

C-----EXTERNAL REFERENCES - NONE
C

C-----CALLED BY - JNUTON, USER ROUTINE
C

C-----COMMONS REFERENCED - JCOEF
C-----VARIABLES -
C     VARIABLE  TYPE      DESCRIPTION
C     -----  -----
C     TJDSTD    R*8       STANDARD EPOCH FOR THE J2000 COORDINATE SYSTEM
C                         IN DAYS PLUS FRACTION OF DAY
C     DAYCEN    R*8       NUMBER OF DAYS IN A JULIAN CENTURY
C     DA1TDT    R*8       A.1 ATOMIC TIME TO TERRESTRIAL DYNAMICAL TIME
C                         (TDT) DIFFERENCE IN DAYS
C     ARCTR     R*8       ARCSECONDS TO RADIAN CONVERSION FACTOR
C     CFOBLM   R*8       COEFFICIENTS FOR THE MEAN OBLIQUITY COMPUTATION
C                         IN ARCSECONDS
C     CFARGS    R*8       COEFFICIENTS FOR COMPUTING THE FUNDAMENTAL
C                         ARGUMENTS OF THE NUTATION SERIES IN ARCSECONDS
C     CFMULT    R*8       COEFFICIENTS OF THE FUNDAMENTAL ARGUMENTS FOR
C                         THE NUTATION SERIES IN ARCSECONDS
C     CFDOBL    R*8       COEFFICIENTS OF THE OBLIQUITY NUTATION TERMS
C                         FOR THE NUTATION SERIES IN ARCSECONDS
C     CFTDOB    R*8       COEFFICIENTS OF THE SECULAR OBLIQUITY NUTATION
C                         TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C     NTERMS    I*4       THE NUMBER OF NONSECULAR TERMS INCLUDED IN THE
C                         COMPUTATION
C     NSECL     I*4       THE NUMBER OF SECULAR TERMS INCLUDED IN THE
C                         COMPUTATION
C

C-----ERROR HANDLING - NONE
C

C-----FILES REFERENCED - NONE
C

C-----DESIGNER - S. DEVLIN, CSC, SEPTEMBER 14, 1987
C

C-----PROGRAMMER - S. DEVLIN, CSC  SEPTEMBER 15, 1987
C

C-----VERIFIED BY -
C

C-----MODIFICATIONS -
C     NAME      DATE      DESCRIPTION
C
CCCC
C     -DECLARE ARGUMENTS
        REAL*8 T2JA1
        REAL*8 AOBL(3,3)
        INTEGER*4 IFDBG
        INTEGER*4 IUDBG
C     -DECLARE LOCAL VARIABLES

```

```

REAL*8 TREF
REAL*8 ARGS(5)
REAL*8 ANGL, DOBLT, OBLTMR, OBLTYR
REAL*8 COBLTY, SOBLTY
INTEGER*4 I, J, IOPT
C
C -COMMON BLOCK JCOEF VARIABLES
REAL*8 TJDSTD
REAL*8 DAYCEN, DA1TDT, ARCTR
REAL*8 CFZETA(6), CFZEE(6), CFTHET(6)
REAL*8 CFOBLM(4)
REAL*8 CFARGS(4,5)
REAL*8 CFMULT(5,35), CFDLON(35), CFDOBL(35)
REAL*8 CFTDLN(1), CFTDOB(1)
INTEGER*4 NTERMS
INTEGER*4 NSECL
C
C COMMON /JCOEF/ TJDSTD, DAYCEN, DA1TDT, ARCTR,
2           CFZETA, CFZEE, CFTHET,
3           CFOBLM, CFARGS, CFMULT, CFDLON, CFDOBL,
4           CFTDLN, CFTDOB, NTERMS, NSECL
C
C
C PRINT DEBUG OUTPUT
IF(IFDBG.GT.0) THEN
  WRITE(IUDBG,1)
  WRITE(IUDBG,1000) T2JA1, IOPT
  IF(IFDBG.GT.1) THEN
    WRITE(IUDBG,2003) TJDSTD, DAYCEN, DA1TDT, ARCTR
  END IF
END IF
C
C COMPUTE THE TIME FROM J2000 TO THE EPOCH TIME IN CENTURIES
TREF = ((T2JA1+DA1TDT) - TJDSTD) / DAYCEN
C
C COMPUTE MEAN OBLIQUITY
C
OBLTMR = ARCTR * (CFOBLM(1) + TREF * (CFOBLM(2) +
2           TREF * (CFOBLM(3) + TREF * CFOBLM(4)) ))
C
IF(IOPT.EQ.1) THEN
  MEAN-OF-DATE
  OBLTYR = OBLTMR
ELSE
  TRUE-OF-DATE
C
C COMPUTE FUNDAMENTAL ARGUMENTS FOR NUTATION SERIES
C
DO 100 I=1,5
  ARGS(I) = ARCTR * (CFARGS(1,I) + TREF * (CFARGS(2,I) +
2           TREF * (CFARGS(3,I) + TREF * CFARGS(4,I)) ))
100 CONTINUE
C
C COMPUTE NUTATION IN OBLIQUITY
C
DOBLT = 0.0D0
DO 300 I=1,NTERMS
C

```

```

      ANGL = 0.0D0
      DO 200 J=1,5
         ANGL = ANGL + CFMULT(J,I) * ARGS(J)
200      CONTINUE
C
      IF(I.GT.NSECL) THEN
         DOBLT = DOBLT + CFDOBL(I) * DCOS(ANGL)
      ELSE
         DOBLT = DOBLT + (CFDOBL(I) + CFTDOB(I)*TREF) * DCOS(ANGL)
      END IF
300      CONTINUE
         DOBLT = ARCTR * DOBLT
C
         OBLTYR = OBLTMR + DOBLT
      END IF
C
C      COMPUTE THE ROTATION MATRIX
C
         COBLTY = DCOS(OBLTYR)
         SOBLTY = DSIN(OBLTYR)
C
         AOBL(1,1) = 1.0D0
         AOBL(1,2) = 0.0D0
         AOBL(1,3) = 0.0D0
         AOBL(2,1) = 0.0D0
         AOBL(2,2) = COBLTY
         AOBL(2,3) = -SOBLTY
         AOBL(3,1) = 0.0D0
         AOBL(3,2) = SOBLTY
         AOBL(3,3) = COBLTY
C
C      PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
         WRITE(IUDBG,2001) TREF, OBLTMR, DOBLT, OBLTYR, ARGS
         WRITE(IUDBG,2002) ((AOBL(I,J),J=1,3),I=1,3)
      END IF
C
1000 FORMAT(' T2JA1 = ',F18.8,' IOPT = ',I4)
2001 FORMAT(
$      ' TREF      = ',D24.15,/,,
$      ' OBLTMR    = ',D24.15,/,,
$      ' DOBLT     = ',D24.15,/,,
$      ' OBLTYR    = ',D24.15,/,,
$      ' ARGS(1)   = ',D22.14,/,,
$      ' ARGS(2)   = ',D22.14,/,,
$      ' ARGS(3)   = ',D22.14,/,,
$      ' ARGS(4)   = ',D22.14,/,,
$      ' ARGS(5)   = ',D22.14)
2002 FORMAT(' AOBL(1,1-3) = ',3F20.15,' AOBL(2,1-3) = ',3F20.15,
2 /' AOBL(3,1-3) = ',3F20.15)
2003 FORMAT(' TJDSTD, DAYCEN, DA1TDT = ',F18.8,F10.2,D24.15)
C
      RETURN
      END
C/ ADD NAME=UTBJCOEF                                04MAY90 15.38.35
C      DATA SET UTBJCOEF    AT LEVEL 003 AS OF 10/30/89
C      DATA SET UTBJCOEF    AT LEVEL 002 AS OF 08/03/88
C      DATA SET UTBJCOEF    AT LEVEL 001 AS OF 07/21/88

```

```

C           DATA SET UTBJCOEF   AT LEVEL 001 AS OF 10/16/87
C           DATA SET BDJCOEF   AT LEVEL 001 AS OF 10/12/87
CCCC
C
BLOCKDATA
C
CX 1 FORMAT(' **** BLOCKDATA BDJCOEF OCTOBER 10, 1987 ****')
C
C-----LANGUAGE - VS FORTRAN
C
C-----FUNCTION - COMMON BLOCK JCOEF CONTAINS COEFFICIENTS FOR THE
C                  PRECESSION COMPUTATION FOR THE MEAN-OF-J2000 COORDINATE
C                  SYSTEM AND COEFFICIENTS FOR THE TRUNCATED VERSION OF
C                  THE NUTATION SERIES COMPUTATION USING THE IAU 1980
C                  THEORY OF NUTATION EQUATIONS FOR THE MEAN-OF-J2000
C                  COORDINATE SYSTEM.
C
C
C-----VARIABLES -
C      VARIABLE  TYPE      DESCRIPTION
C      -----  -----
C      TJDSTD    R*8      STANDARD EPOCH FOR THE J2000 COORDINATE SYSTEM
C                          IN DAYS PLUS FRACTION OF DAY
C      DAYCEN    R*8      NUMBER OF DAYS IN A JULIAN CENTURY
C      DA1TDT    R*8      A.1 ATOMIC TIME TO TERRESTRIAL DYNAMICAL TIME
C                          (TDT) DIFFERENCE IN DAYS
C                          ( (32.184D0-0.0343817D0) / 86400.0D0 )
C      ARCTR     R*8      ARCSECONDS TO RADIAN CONVERSION FACTOR
C                          (0.0174532925199433D0 / 3600.D0)
C      CFZETA    R*8      COEFFICIENTS FOR PRECESSION ANGLE ZETA
C                          IN ARCSECONDS
C      CFZEE     R*8      COEFFICIENTS FOR PRECESSION ANGLE Z
C                          IN ARCSECONDS
C      CFTHET    R*8      COEFFICIENTS FOR PRECESSION ANGLE THETA
C                          IN ARCSECONDS
C      CFOBLM    R*8      COEFFICIENTS FOR THE MEAN OBLIQUITY COMPUTATION
C                          IN ARCSECONDS
C      CFARGS    R*8      COEFFICIENTS FOR COMPUTING THE FUNDAMENTAL
C                          ARGUMENTS OF THE NUTATION SERIES IN ARCSECONDS
C      CFMULT    R*8      COEFFICIENTS OF THE FUNDAMENTAL ARGUMENTS FOR
C                          THE NUTATION SERIES IN ARCSECONDS
C      CFDLON    R*8      COEFFICIENTS OF THE LONGITUDE NUTATION TERMS
C                          FOR THE NUTATION SERIES IN ARCSECONDS
C      CFDOBL    R*8      COEFFICIENTS OF THE OBLIQUITY NUTATION TERMS
C                          FOR THE NUTATION SERIES IN ARCSECONDS
C      CFTDLN    R*8      COEFFICIENTS OF THE SECULAR LONGITUDE NUTATION
C                          TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C      CFTDOB    R*8      COEFFICIENTS OF THE SECULAR OBLIQUITY NUTATION
C                          TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C      NTERMS    I*4      THE NUMBER OF NONSECULAR TERMS INCLUDED IN THE
C                          COMPUTATION
C      NSECL     I*4      THE NUMBER OF SECULAR TERMS INCLUDED IN THE
C                          COMPUTATION
C
C-----DESIGNER - S. DEVLIN, CSC, SEPTEMBER 14, 1987
C
C-----PROGRAMMER - S. DEVLIN, CSC, SEPTEMBER 15, 1987
C

```

```

C-----VERIFIED BY -
C
C-----MODIFICATIONS -
C      NAME          DATE      DESCRIPTION
C      S. DEVLIN    10/12/87  ADD D0 TO CFMULT ARRAY VALUES
C
CCCC
C      -COMMON BLOCK JCOEF VARIABLES
      REAL*8 TJDSTD
      REAL*8 DAYCEN, DA1TDT, ARCTR
      REAL*8 CFZETA(6), CFZEE(6), CFTHET(6)
      REAL*8 CFOBLM(4)
      REAL*8 CFARGS(4,5)
      REAL*8 CFMULT(5,35), CFDLON(35), CFDOBL(35)
      REAL*8 CFTDLN(1), CFTDOB(1)
      INTEGER*4 NTERMS
      INTEGER*4 NSECL

C
      COMMON /JCOEF/ TJDSTD, DAYCEN, DA1TDT, ARCTR,
2                  CFZETA, CFZEE, CFTHET,
3                  CFOBLM, CFARGS, CFMULT, CFDLON, CFDOBL,
4                  CFTDLN, CFTDOB, NTERMS, NSECL

C
      DATA TJDSTD/2451545.0D0/
      DATA DAYCEN/36525.0D0/
      DATA DA1TDT/0.372102063657407D-03/
      DATA ARCTR/0.484813681109536D-05/
      DATA CFZETA /2306.2181D0,   1.39656D0, -0.000139D0,
2                 0.30188D0, -0.000344D0,  0.017998D0/
      DATA CFZEE  /2306.2181D0,   1.39656D0, -0.000139D0,
2                 1.09468D0,  0.000066D0,  0.018203D0/
      DATA CFTHET /2004.3109D0, -0.85330D0, -0.000217D0,
2                 -0.42665D0, -0.000217D0, -0.041833D0/
      DATA CFOBLM/84381.448D0, -46.8150D0, -0.00059D0,  0.001813D0/
      DATA CFARGS/ 485866.733D0, 1717915922.633D0, 31.310D0,  0.064D0,
2                 1287099.804D0, 129596581.224D0, -0.577D0, -0.012D0,
3                 335778.877D0, 1739527263.137D0, -13.257D0,  0.011D0,
4                 1072261.307D0, 1602961601.328D0, -6.891D0,  0.019D0,
4                 450160.280D0, -6962890.539D0,  7.455D0,  0.008D0/
      DATA CFMULT/0.D0,0.D0,0.D0,0.D0,1.D0,  0.D0,0.D0,0.D0,0.D0,2.D0,
2                 -2.D0,0.D0,2.D0,0.D0,1.D0,  2.D0,0.D0,-2.D0,0.D0,0.D0,
3                 0.D0,0.D0,2.D0,-2.D0,2.D0,  0.D0,1.D0,0.D0,0.D0,0.D0,
4                 0.D0,1.D0,2.D0,-2.D0,2.D0,  0.D0,-1.D0,2.D0,-2.D0,2.D0,
5                 0.D0,0.D0,2.D0,-2.D0,1.D0,  2.D0,0.D0,0.D0,-2.D0,0.D0,
6                 0.D0,0.D0,2.D0,-2.D0,0.D0,  0.D0,2.D0,0.D0,0.D0,0.D0,
7                 0.D0,1.D0,0.D0,0.D0,1.D0,  0.D0,2.D0,2.D0,-2.D0,2.D0,
8                 0.D0,-1.D0,0.D0,0.D0,1.D0,  0.D0,0.D0,2.D0,0.D0,2.D0,
9                 1.D0,0.D0,0.D0,0.D0,0.D0,  0.D0,0.D0,2.D0,0.D0,1.D0,
A                 1.D0,0.D0,2.D0,0.D0,2.D0,  1.D0,0.D0,0.D0,-2.D0,0.D0,
B                 -1.D0,0.D0,2.D0,0.D0,2.D0,  0.D0,0.D0,0.D0,2.D0,0.D0,
C                 1.D0,0.D0,0.D0,0.D0,1.D0, -1.D0,0.D0,0.D0,0.D0,1.D0,
D                 -1.D0,0.D0,2.D0,2.D0,2.D0,  1.D0,0.D0,2.D0,0.D0,1.D0,
E                 0.D0,0.D0,2.D0,2.D0,2.D0,  2.D0,0.D0,0.D0,0.D0,0.D0,
F                 1.D0,0.D0,2.D0,-2.D0,2.D0,  2.D0,0.D0,2.D0,0.D0,2.D0,
G                 0.D0,0.D0,2.D0,0.D0,0.D0, -1.D0,0.D0,2.D0,0.D0,1.D0,
H                 -1.D0,0.D0,0.D0,2.D0,1.D0,  1.D0,0.D0,0.D0,-2.D0,1.D0,
I                 -1.D0,0.D0,2.D0,2.D0,1.D0/
      DATA CFDLON/-17.1996D0,  0.2062D0, 0.0046D0,  0.0011D0, -1.3187D0,

```

```
2          0.1426D0, -0.0517D0, 0.0217D0, 0.0129D0, 0.0048D0,
3          -0.0022D0, 0.0017D0, -0.0015D0, -0.0016D0, -0.0012D0,
4          -0.2274D0, 0.0712D0, -0.0386D0, -0.0301D0, -0.0158D0,
5          0.0123D0, 0.0063D0, 0.0063D0, -0.0058D0, -0.0059D0,
6          -0.0051D0, -0.0038D0, 0.0029D0, 0.0029D0, -0.0031D0,
7          0.0026D0, 0.0021D0, 0.0016D0, -0.0013D0, -0.0010D0/
DATA CFDOBL/9.2025D0, -0.0895D0, -0.0024D0, 0.D0, 0.5736D0,
2          0.0054D0, 0.0224D0, -0.0095D0, -0.0070D0, 0.0001D0,
3          0.0D0, 0.0D0, 0.0009D0, 0.0007D0, 0.0006D0,
4          0.0977D0, -0.0007D0, 0.0200D0, 0.0129D0, -0.0001D0,
5          -0.0053D0, -0.0002D0, -0.0033D0, 0.0032D0, 0.0026D0,
6          0.0027D0, 0.0016D0, -0.0001D0, -0.0012D0, 0.0013D0,
7          -0.0001D0, -0.0010D0, -0.0008D0, 0.0007D0, 0.0005D0/
DATA CFTDLN/-0.01742D0/
DATA CFTDOB/0.00089D0/
DATA NTERMS/35/
DATA NSECL/1/
C
END
```